

## NIRMA UNIVERSITY

<b>Institute:</b>	Institute of Technology
<b>Name of Programme:</b>	BTech CSE, Integrated BTech (CSE)-MBA, BTech CSE (Artificial Intelligence & Machine Learning)
<b>Course Code:</b>	XXXX
<b>Course Title:</b>	Advanced Data Structures
<b>Course Type:</b>	Department Elective-II
<b>Year of Introduction:</b>	2024-25

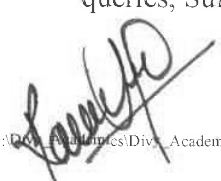
L	T	Practical Component				C
		LPW	PW	W	S	
3	0	2	-	-	-	4

### Course Learning Outcomes (CLO):

At the end of the course, the students will be able to –

1. interpret the trade-offs involved in choosing between different data structures (BL2)
2. apply advanced data structures to solve real-world problems (BL3)
3. analyse the time and space complexity of algorithms (BL4)
4. design and implement advanced data structures. (BL6)

Unit	Contents	Teaching Hours (Total 45)
Unit-I	<b>Search Trees:</b> Models of Search Trees, Properties and transformations, height of search tree, basic find, insert and delete, returning from leaf to root, dealing with non-unique keys, queries for keys in an interval, building optimal search trees, converting trees to lists, removing a tree.	09
Unit-II	<b>Balanced Search Trees:</b> Height-balanced and weight-balanced trees, B-trees, Red Black Trees and Trees of almost optimal height, Finger trees and level linking, trees with partial rebuilding, Splay Trees, Skip Trees, Joining and Splitting Balanced Search Trees.	09
Unit-III	<b>Tree Search for Set of Intervals:</b> Interval Trees, Trees for the union of intervals, trees for sums of weighted intervals, trees for interval-restricted maximum sum queries, orthogonal range trees, higher dimensional segment trees, other systems of building blocks, range counting and semigroup model, Quad-tree, kd-trees and related structures.	09
Unit-IV	<b>Heaps:</b> Array-based heaps, heap-ordered trees and half-ordered trees, Leftist Heaps, Skew heaps, Binomial heaps, changing keys in heaps, Fibonacci heaps, heaps of optimal complexity, Double ended heap structures and multidimensional heaps, heap-related structures with constant time updates.	09
Unit-V	<b>Union – Find and related structures:</b> Union – Find, Union Find with copies and dynamic segment trees, list splitting, Problems on root-directed trees, maintaining a linear order Data Structure Transformations: Making structures dynamic and persistent	06
Unit-VI	<b>Hashing and dictionary operations:</b> Static & Dynamic Hashing techniques, Tries and compressed tries, Dictionaries allowing errors in queries, Suffix Trees, Suffix arrays	03



**Self-Study:**

The self-study contents will be declared at the commencement of the semester. Around 10% of the questions will be asked from self-study content

**Suggested Readings/ References:**

1. Peter Brass, Advanced Data Structures, Cambridge University Press
2. Suman Saha, S. Shukla, Advanced Data Structures: Theory and application, CRC press
3. A.A. Puntambekar, Advanced Data Structures, Technical Publications

**Laboratory Work:**

Laboratory work will be based on the above syllabus with a minimum of 10 experiments to be incorporated. The students in a suitable group size will design and perform one experiment as a part of Laboratory work.

Sr. No.	List of Experiments/Exercises	Hours																						
1	Implement the shadow copying technique for the STACK data structure to solve the MAXZISE problem.	02																						
2	Implement a balanced binary search tree (AVL). The search tree operations, create, insert, delete, and display, should be included. The input for creating the tree should be taken from a text/CSV file. The content of the file should be a unique key-object pair.	04																						
3	Rebalancing operations can be delayed until a certain threshold is attained. The scapegoat tree uses partial rebuilding to balance a search tree. Implement a scapegoat tree to demonstrate the partial rebuilding operation.	04																						
4	Skip list structures are used to retrieve the data faster. Implement the structure up to the third level. Show the effect of the insert and delete operations.	02																						
5	Write a program to split a balanced search tree at i. Root ii. A given point of split.	04																						
6	Segment trees are useful for finding the range sum of a given interval. Write a program to demonstrate the usage of the segment tree structure to find the range sum of numbers in a given range. Example: Given	02																						
	<table border="1"> <tbody> <tr> <td>Index</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> </tr> <tr> <td>Data</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> <td>10</td> <td>11</td> </tr> </tbody> </table>	Index	0	1	2	3	4	5	6	7	8	9	Data	2	3	4	5	6	7	8	9	10	11	
Index	0	1	2	3	4	5	6	7	8	9														
Data	2	3	4	5	6	7	8	9	10	11														
	Sum (0,4) = 20 Sum (2,6) = 30																							
7	Implement a heap data structure using a linked list structure. The list should retrieve high-priority objects every time the extract operation is performed.	04																						
8	Write a program to implement a union-find structure. The program should demonstrate the structure representation of the set and list the items of the selected set.	04																						
9	Suffix arrays are preprocessed structures that can be used to solve the classical substring matching problem. Implement suffix arrays for a long string sequence and demonstrate the matching operation.	02																						
10	Hash tables are important data structures. However, hash tables are subject to collision. Implement a program with a collision resolution technique with Insert, delete, and display operations	02																						